Authors:	Keskustalo Heikki, Järvelin Kalervo
Name of article:	Simulations as a Means to Address Some Limitations of Laboratory-based IR Evaluation
Name of work:	The Janus Faced Scholar: A Festschrift in Honour of Peter Ingwersen
Editors of work:	Larsen Birger et al
Year of publication:	2010
ISBN:	978-87-741-5318-4
Pages:	69-86
Series name and number:	Special volume of the e-zine of the international society for scientometrics and informetrics
Discipline:	Natural sciences / Computer and information sciences
Language:	en
School/Other Unit:	School of Information Sciences

URL: http://www.issi-society.info/peteringwersen/pif_online.pdf
URN: http://urn.fi/urn:nbn:uta-3-475

Preprint from: Keskustalo, H. & Järvelin, K. (2010). Simulations as a Means to Address Some Limitations of Laboratory-based IR Evaluation. In: Larsen, B. & al. (Eds.). The Janus Faced Scholar: A Festschrift in Honour of Peter Ingwersen. A Special volume of the e-zine of International Society for Scientometrics and Informetrics, vol. 06-S June 2010. pp. 69-86.

Simulations as a Means to Address Some Limitations of Laboratory-based IR Evaluation

Heikki Keskustalo, Kalervo Järvelin

Department of Information Studies and Interactive Media, FI-33014 University of Tampere, FINLAND
{Heikki.Keskustalo, Kalervo.Jarvelin}@uta.fi

Abstract. We suggest using simulations to address some of the limitations of test collection-based IR evaluation. In the present paper we explore the effectiveness of short query sessions based on a graph-based view of the searching situation where potential queries (query key combinations) constitute the vertexes of a graph G describing each topic. “Session strategies” are rules which determine the acceptable query reformulations. Query reformulations manifest as edges in G, and they express the allowed transitions between the vertexes. Multiple-query topical sessions manifest as paths in G. We present an example of this approach assuming session strategies based on limited query modifications (additions, deletions, or substitutions of few query words). We end by discussing the significance of our approach for IR evaluation.

1 Introduction

In their seminal book *The Turn* Ingwersen and Järvelin point out some of the main problems related to the laboratory-based IR evaluation, including the lack of modeling explicit users and tasks, and the lack of modeling interaction ([1]; see [2] for the original discussion). Recent studies suggest that in real life users typically prefer short queries, try out more than one query if needed [3-7] and often prefer making only small modifications to their queries [3]. Furthermore, even experts encountering the same task may use very different wordings in their searching. They may also consider finding only a few reasonably good documents as success [4]. Users also try to compensate for the performance deficiencies of the systems by adapting their search behavior [5, 7, 8]. The traditional Cranfield-style experiments based on one query per topic are not well-suited to study such behavior.

We suggest using simulations as a solution towards some of the limitations of Cranfield-style experiments discussed in Turn. By simulations we refer to experimentation based on using a symbolic model of a simplified real life search sessions in order to answer research questions. We assume multiple-query search sessions based on alternating querying and browsing phases. In the present paper, in particular, we will simulate search sessions assuming the shortest queries (including several one-word query versions for every topic). We allow several queries for a topic, assume limited modifications to the queries, and define success as being able to find one (highly) relevant document for a topic.

In other words, we restrict our attention to a simulation where short queries are used in various combinations in sessions. We assume that the searcher issues an initial query and inspects some top-N documents retrieved; if an insufficient number of relevant documents are recognized, the user repeatedly launches queries until the information need is satisfied or the user gives up.

The motivation behind our approach is that due to the costs involved during query formulation, the user may optimize the total cost-and-benefit of his sessions by rapidly trying out short queries. In other words, the user is willing to take chances with the quality of the result, and he is prepared to try out several short queries to see if something relevant is to be found.

Formally characterized, we utilize a graph-based approach in test collections explained in Section 3. In the experimental part of the study we will utilize the TREC 7-8 corpus with 41 topics having graded relevance assessments.

Next we will briefly review literature on user behavior and justify our approach. This is followed by defining our research problem. Section 3 explains the graph-based simulation method. Results of our experiments are given in Section 4. Discussion and conclusions are presented in Section 5.

2. The Significance of Multiple-Query Sessions

2.1 User behavior

Searchers behave individually in real life: their information needs may be unclear and dynamic as the users may learn as the session progresses, and the users may switch focus. In practice, a particular searcher may try out several queries during a search session, and different searchers may try out different wordings even when they face the same (well-defined) search task. It may be difficult for the searcher to predict how well a particular query will perform [8] because even assuming that the query does describe the topic well, it may be ambiguous [9] and therefore not retrieve documents serving the particular searcher in his searching context. Therefore, multiple query sessions are commonplace and may be unavoidable in practice in real life.

It has also been observed that real searchers often make use of very short queries and they prefer making small modifications to the previous queries. Jansen and colleagues [3] analyzed transaction logs containing thousands of queries posed by Internet search service users. They discovered that one in three queries had only one term; two in three had one or two terms. On the average the query length was 2.21 terms per query. The average number of terms used in a query was even smaller, 1.45, in a study by [6] focusing on intranet users. Less than 4 % of the queries in Jansen's study had more than 6 terms. Because very short queries are commonplace, focusing on them in a test collection environment study seems justified.

Real-life searchers also avoid excessive browsing. They may stop browsing if the search result does not look promising almost immediately [10]. The stopping decisions regarding browsing the retrieved document list depend on the search task and the individual [4]. Jansen and colleagues [3] observed that most users did not access results past the first page presenting the top-10 results retrieved. Users may stop the search session after finding one or a few relevant documents. In particular, real searchers very rarely browse the top-1000 documents, although in some cases they do (e.g., patent searchers). Therefore, it is important to study situations where the search is successfully completed after only one or few relevant documents are found.

2.2 Motivation and research question

Generally speaking, valid instruments and study designs used to explain or evaluate some phenomenon should incorporate major factors affecting the phenomenon under study and systematically relate them to each other. We justify our present study design by the following observations. First, in real life users often:

- prefer very short queries (often only 1-2 keys)
- try out more than one query per topic, if needed
- cope by trying out limited modifications to queries
- avoid browsing a long list of documents, and
- stop after finding one or a few relevant documents

In traditional Cranfield-style experiments, it is common to (implicitly) assume fundamentally different kind of user behavior. These studies are typically based on using:

- longer queries (at least somewhat longer, e.g., even title queries typically have more than one word)
- one query per topic (and presenting the results averaged over topics)

Therefore, in the present paper we suggest modeling user behavior, in a test collection, but using:

- the very short queries
- several queries per topic
- limited word-level edit operations to modify queries
- shallow browsing, and
- one or a few (highly) relevant documents as the success criterion

Regarding the first two items, we will construct several alternative one-word query candidates, and slightly longer queries, for each topic. One way to approach searching is to use one-word queries as the starting points for sessions. Regarding the third item we assume that queries are modified by performing limited word additions, deletions, or substitutions.

Regarding the last two items, we assume that if any particular query within a session fails, the user will stop browsing almost immediately (N.B., this makes sense because the simulated user is aware that a short query attempt may very well fail).

If a query is successful, the user will stop searching after finding one (highly) relevant document. We use precision at 5 documents (P@5) as our primary success criterion and experiment with two separate relevance thresholds – liberal and stringent (see Section 2.3) [11].

A successful end result for any search session may require a different number of queries for individual topics. For one topic the first query candidate may be successful – as we will show - while for the next topic additional query candidates may be required.

Research question

Our overall research question in this paper is: *How successful are short queries as sessions when we assume limited query modifications, limited browsing and success defined as being able to find one (highly) relevant document?*

In studying this problem, we will assume that:

- the topical requests remain unchanged during a session - the simulated searcher neither learns nor switches focus during the session;
- the relevance of the documents for the simulated searcher is defined by the recall base of the test collection; and
- the simulated searcher scans the ranked list of documents from the top to bottom – behavior observed via eye-tracking [12].

2.3 The test collection and search engine

We used the reassessed TREC test collection including 41 topics from TREC 7 and TREC 8 ad hoc tracks [11]. The document database contains 528155 documents organized under the retrieval system Lemur. The relevance judgments are done on a four-point scale: (0) irrelevant; (1) marginally relevant: the document only points to the topic but does not contain more or other information than the topic description; (2) fairly relevant: the document contains more information than the topic description but the presentation is not exhaustive; and (3) highly relevant: the document discusses the themes of the topic exhaustively. In the recall base there are on the average 29 marginally relevant documents, 20 fairly relevant documents and 10 highly relevant documents for each topic [11].

2.4 Collecting the query data

All test topics were first analyzed intellectually by two sets of test persons to form query candidate sets. Our intention was to collect a reasonable set of query candidates together with user estimations regarding their appropriateness. During the topic analysis the test persons did not interact with a real system. They probably would have been able to make higher quality queries, if they had had a chance to utilize system feedback. However, this is no limitation to the method described in this paper. We demonstrate here our graph-based method based on data collected from a group of seven undergraduate information science students. Regarding each topic a printed topic description and a task questionnaire were presented to the test persons. Each person analyzed six topics (one person analyzed five topics) thus 41 topics were analyzed. The users were asked to directly select and to think up good search words from topical descriptions; to create various query candidates; and to evaluate how appropriate the query candidates were.

The test persons were asked to form query versions of various lengths. We used the long query version requested to have three or more words as a starting point: first we selected its first three words A-C for each topic. To get the needed fourth and the fifth word we selected randomly distinct words from the remaining words in the long query version, or, if its words run out, from the other query versions requested from the users. Our goal in using the data collected from the test persons was to define a set of five query words for each topic. The procedure produced some obvious bad keys for topics (see Appendix) but this only makes our argument stronger - if the empirical results show that as sessions these words, tried as various combinations, often produce a rapid success despite some bad keys included.

3. Graph-Based Simulation

Our suggested procedure described next is inspired mainly by two main points: (1) real users cope with short queries, and (2) they prefer small query modification steps. In brief, our graph-based method to study multiple-query session effectiveness in a test collection consists of the following steps:

1. Words are collected to describe the test topics. Sources of data include using topic descriptions of test collections directly; utilizing test persons performing simulated or real tasks, etc. We asked test persons to create realistic content for short topical queries.
2. Query candidates are formed for each topic. We formed all possible word combinations (of 5 word) using the bag of words operator #sum of Lemur. However, queries may have some other structure, e.g., the #and or proximity operators. The basic idea is to create an extensive listing of possible query types (cf. [13]).
3. A search is performed using each query combination for each topic. We used the Lemur retrieval system in our experiment producing a ranked list of retrieved document, but other types of retrieval engines, e.g., Boolean systems, could be utilized.
4. Each distinct query is interpreted as a vertex of a (topical) graph.
5. The effectiveness results (regarding each distinct query) are expressed alongside the vertexes.
6. Sessions are now considered - in retrospect. To do this, we study the properties of the graphs.

To simulate sessions we need to (1) select start vertex; (2) determine the traversal rule(s); (3) define the stopping condition(s), and (4) consider the vertex traversal for each topic. For example

- One-word queries may be considered as start vertexes.
- "One word can be added/deleted/substituted at time" is one example of a traversal rule (a query modification rule).
- "Stop if 1 highly relevant document is found" is an example of a stopping condition.

7. The properties of sessions (paths) can be studied by using various effectiveness metrics.

If all word combinations are formed, their number increases rapidly as the number of keys increases. We limit our experiment to 5 query keys for each topic thus producing 25 graph vertexes.

Vertexes of the graph

In more detail, the simulation process goes as follows. First, the set of vertexes is formed for each topic. We assume unstructured *(#sum)* queries. Each distinct query (query key combination) constitutes one vertex $v_i \in V$ in a directed acyclic graph $G = (V, E)$. The query reformulations are reflected as edges ($e_j \in E$) in G and they express the allowed transitions between the vertexes. Multiple-query topical sessions manifest as paths in *G*. We have an ordered list of 5 query keys A, B, C, D, E available for each topic in our test data. These five keys produce 25 query combinations. In other words, 32 vertexes of the (topical) query graph are created (31 vertexes if the empty query is excluded). The vertexes are arranged in Table 1 into a diamond-shaped figure so that the number of keys increases in the query combinations from top to bottom.

Table 1. Query combinations (graph vertexes) arranged by the number of keys.

{}									
A	B	C	D	E					
AB	AC	AD	AE	BC	BD	BE	CD	CE	DE
ABC	ABD	ABE	ACD	ACE	ADE	BCD	BCE	BDE	CDE
ABCD	ABCE	ABDE	ACDE	BCDE					
ABCDE									

The figure consists of 6 rows - from top to bottom - one empty query vertex; 5 one-word vertexes; 10 two-word vertexes; 10 three-word vertexes; 5 four-word vertexes and one 5-word vertex. Top-1000 documents are retrieved using each query. For each individual topic the diamond-shaped graph below is formed, and the selected effectiveness values are computed for each vertex. Also the corresponding average figures over 41 topics (liberal relevance threshold) or a subset of 38 topics (stringent relevance threshold) may be computed.

For example, assuming an ordered list of individual query keys A, B, C, D, E, the vertex BC is used to denote an (unstructured) two-word query consisting of the second and the third query key. For example, the query keys A-E for topic #351 constitute an ordered set {petroleum, exploration, south, atlantic, falkland} (see Appendix). In this case the vertex BC corresponds to the query #sum(exploration south).

Edges of the graph

Based on literature, we hypothesize that topical query sessions are often constituted by implicit / educated / learned "moves" between the vertexes. Obviously, the user has to start somehow. We assume that the user proceeds from one vertex and moves into another (creating a directed edge) by applying some acceptable (albeit implicit) rules or heuristics. One such possible user rule would be – based on the principle of least effort - to allow word-edit operations that have a cost of one - compared to the previous query. Such a user would add, delete or edit one word compared to the previous query formulation. In other words, the user tries to cope with a situation by making small, incremental steps.

Session strategies

The success of various query sequences as topical sessions may be analyzed in relation to the start vertexes, the traversal rules, and the stopping condition:

- *Selection of the start vertex.* The effects of selecting the start vertex from some particular level of the graph may be immediately inspected.
- *Traversal rules.* We restrict our attention to consider traversal rules based on small modifications. According to [3] modifications to successive queries are done in small increments; it is common to modify, add or delete a search key.
- *Stopping condition.* As explained, in the present paper we consider the task of finding one (highly) relevant document.

Regarding the graph, we know the exact form of the query in each node (both the "identity" and number of words in it), and its success (measured, e.g., as P@5 using the stringent relevance threshold). We can perform retrospective analyses regarding query sessions after defining the traversal rules (how to move from one node to another) and the stopping condition (what constitutes success). Our purpose is to consider the concept of a session using the data in the graph in retrospect. The vertexes allow us to see what would happen assuming various session strategies and criteria for session success. The graph gives an overview of success assuming different types of queries (e.g., several alternative one-word queries).

4. Results

Next we will discuss three kinds of results. First, we show general results for P@5 values (averaged over topics) using two relevance thresholds (Tables 2-3). The cells in the figure correspond to the query combinations explicated in Table 1. Second, we concentrate on the case of highly relevant documents required. Table 4 shows the share of successful topics, i.e., when a particular query combination was successful in finding a highly relevant document in the top-5.

Last, we will study how successful small query modifications are within sessions (if the current query fails). This analysis needs to be performed topic by topic. Therefore, we first illustrate the results for one topic (Table 5), present the data as a binary phenomenon, and finally present session information for all topics as a binary map (Table 6)

Liberal relevance threshold

In Table 2 following general trend emerges: P@5 gets higher values when we move downwards (i.e., towards the longer queries) and towards left in the graph. On the one hand, it seems that our one-word queries were a “bad call”, because even in the best case (the first individual word selected for each topic) the P@5 figure is low (13.7 %). On the other hand it seems that we selected the query keys in the correct order: the first single words selected (the left-most keys) are, on the average, more successful than the last words (P@5 figure 4.9 % for the 5th individual keys). We next repeat the previous experiment but this time accepting only the highly relevant documents as success (Table 3).

Table 2. Effectiveness (P@5) (%) averaged over topics (N=41) for the various query combinations (liberal relevance threshold). See Table 1 for the queries in each cell.

-									
13.7	13.2	3.9	3.9	4.9					
34.2	27.8	26.8	24.4	21.5	19.5	17.1	9.8	10.7	6.8
44.9	40.0	40.1	36.6	34.2	29.3	31.7	27.3	22.9	11.2
48.3	43.9	44.4	35.1	28.8					
49.8									

Stringent relevance threshold

In Table 3 the same kind of pattern as in Table 2, only weaker, emerges. Again, obviously, basically it seems that we can state, regarding the query length, “the longer the better”. Yet, a problem with the numbers in Tables 2 and 3 is that they are impossible to interpret regarding individual topical sessions. Because of this, we will next look at the number of topics for which (at top-5 documents) the queries succeeded. We count the share of topics, out of 38, for which at least one highly relevant document was found in top-5.

Table 3. Effectiveness (P@5) (%) averaged over topics (N=38) for various queries (stringent relevance threshold).

-									
7.4	6.8	0.5	1.1	1.6					
13.7	12.1	11.1	10.5	4.7	9.0	7.4	4.2	4.7	2.1
15.8	14.7	16.3	16.8	16.3	10.5	11.6	7.9	10.5	6.3
17.9	16.3	17.9	15.8	11.1					
19.5									

Failures become rarer as the queries get longer. This happens rapidly: by using two reasonable keys (e.g., any one of the combinations AB, AC, and AD) the user succeeds for slightly less than half of the topics (failures for 21, 24, and 22 topics corresponding to success in case of 45 %, 37 %, and 42 % of the topics). Interestingly, the distinction between the best 3-word and 4-word queries seems to disappear measured this way, and they are almost as successful as the 5-word queries. We would like to draw the attention of the reader to the fact that it is not possible to interpret the data in Table 4 much more deeply without considering queries as sequences, and regarding individual topics. For example, one may claim that queries of type E are generally inferior compared to the queries of type A. While this indeed is true, e.g., for the individual topic #351 query A fails but query E succeeds. Also in real life sometimes a (short) query succeeds, sometimes it fails. In that case the user may start reformulating queries. We will next enter into this territory through retrospective session analysis.

Table 4. The share of successful topics (%) for which at least one highly relevant document was retrieved at top-5. N=38 topics.

-									
24	21	3	5	8					
45	37	42	34	21	26	26	18	16	8
55	53	53	47	53	37	39	26	37	21
55	53	53	53	37					
61									

Sessions are next considered as traversals (paths) where the user continues the topical session and launches the next query if and only if any current query fails. We start by showing how to present the success of the component queries for one topic (#351).

Individual query example

Our analysis is limited by the assumption that the user considers only the set of words (5 in our case) available. Although we limit our experiments to 5 words, larger word sets could be used. However, it is not unrealistic to assume that a user may cope in a retrieval situation by indeed using a limited set of query keys. As our results show, if the user is able to invent one or two good keys, (s)he may succeed.

Table 5. Effectiveness (P@5) (%) for topic #351 ("petroleum exploration south atlantic falkland") measured at stringent relevance threshold, for various query combinations. 14 highly relevant documents exist for the topic. Legend: cells with a value above zero indicate success (+) and zeros indicate failure (-) for any particular query combination.

-									
0	0	0	0	0.2					
0	0	0	0.4	0	0	0.4	0	0	0
0	0	0.4	0	0.2	0.2	0	0.2	0.2	0
0	0.4	0.4	0.2	0.2					
0.4									

Table 5 allows studying, in retrospect, the effects of using various session approaches. We may analyze the general level of success through the number of words in queries and traversals via word-level substitution, addition and deletion.

Binary session map

Numbers in the graph vertexes in Table 5 can be interpreted as binary success (e.g., when at least one highly relevant document is found within the top-5, i.e., P@5>0) or failure (otherwise). By labeling the successful vertexes by a plus ('+') sign and the failed vertexes by a minus ('-') sign, information in Table 5 can be expressed in form of a character string:

```
#351 ----+ ---+--+--- --+-++-++- -++++ +
```

To make the diagram readable we arranged it into groups of 5, 10, 10, 5, and 1 symbol, corresponding to query combinations having one, two, three, four, and five query keys. By expressing the topical data this way for every topic a visual map is created. It gives information regarding the query combinations available for topical sessions based on a specific success criterion (Table 6).

Table 6. Binary session map for 38 topics and all query combinations. Legend: plus ('+') or minus ('-') symbols correspond to the 31 non-empty vertexes in the topical graph, traversed left to right, and rows traversed from top to bottom. Plus indicates success, i.e., P@5 > 0 (stringent relevance threshold) and minus indicates a failure (P@5 = 0).

```
#351 ----+ ---+--+--- --+-++-++- -++++ +
#353 ----- --------+- ----+----+ ---+- -
#355 +++-+ ++++++++++ ++++++++++ +++++ +
#358 ----- -+++---+-- ---+++---+ ---+- +
#360 -+--- +----++--- -++-----+- --+-- -
#362 ----- --------+- ---------- ----- -
#364 +---- ++++------ ++++++---- ++++- +
#365 -+--- +-+-+++--- +++++++++- +++++ +
#372 ----- -+--+--+-- +--++-+--+ ++-++ +
#373 +---- -+++------ +--+++---- ++-+- -
#377 -+--- +---+++--- +++---+++- +++-+ +
#384 ----- -----+-+-- -+-+--+-+- +-+++ +
#385 ----- ++++------ +++++++-+- +++++ +
#387 -+--- +-++-++--- +++++++++- +++-- +
#388 ----- ---------- ---------- ----- -
#392 -+--- ------+--- --+----++- ----- -
#393 ---++ ++++-+++++ ++++++++++ +++++ +
#396 -+-+- -++--+++-- ++++++--++ ++++- +
#399 ----- ----+----- ---------- ----- -
#400 ++--- ++-------- -++++----- ---+- -
#402 ----- ---------- ---------- ----- -
#403 ----- +-+++----- +++++-++-+ ++-++ +
#405 ----- ---------- --+------- -++-- +
#407 +---- ++++---++- ++++++++-- ++++- +
#408 ----- ---------- ------+--- ----+ +
#410 +---- +++------- +++++----- ++++- +
#415 ----- ---------- ++----+--- +-+-+ +
#416 +---- -+++------ ++-+------ +-++- +
#418 +---- ++++------ ++++++---- ++++- +
#420 ----- -++++++--- +++++++++- +++++ +
#421 ----- +--------- +--------- ----- -
#427 ----- --------++ ------+-++ ---++ -
#428 ----- +---+----- +--------- ++--- -
#431 +---- +-+------- +++-++---- +++-- +
#440 ----- ---------- ----+----- ----- -
#442 ----- ---------- ---------- ----- -
#445 ----- +----+---- +++---+-+- +++-+ +
#448 ----- ---------- ---------- ----- -
```

In Table 6 the very first symbols of each group are especially interesting. For example, the first symbols of the first three groups represent, correspondingly, the queries of type A, AB, ABC. As the test persons were requested to express each topic by using three or more words, these three query types are formed from the very first words (left to right) as listed by the test persons. We will next briefly discuss the properties of one to three word queries in sessions.

One-word queries

Table 6 shows the success of one-word queries (the first group of five symbols in each line) in sessions. We can see that the very first single-word query ('A') succeeded for 9 topics (#355, #364, #373, …) (the first symbol of the first group). Assuming that the user started the session this way and in case of failure continued by trying out the second single-word query ('B')(substitution of the key), it succeeded for 6 additional topics (#360, #365, #377, …) (the second symbol of the first group). Assuming, that the user continued instead by adding one word ('AB'), it succeeded even better, for 10 additional topics (#360, #365, #377, …)(the first symbol of the second group). Obviously, there are limits for this one-word approach as in case of 17 topics out of 38 at least one of the one-word queries succeeded.

Two-word queries

If the session was started by trying out a two-word query (the first two words given by the simulated users: 'AB') it succeeds for 17 topics (#355, #360, #364, …) out of 38. Assuming that the user continues in case of failure by trying out the second two-word query ('AC')(substitution of the second query key), it succeeds for 6 additional topics (#358, #372, #373, …). For 21 topics every one-word query failed, but a successful two-word query can be found for these in 13 cases (#353, #358, #362, …).

Three-word queries

If the session was started by a three-word query (the first three words given by the simulated users: 'ABC') the session immediately succeeds for 21 topics (#355, #364, #365, …) out of 38. Assuming that the user continues, in case of failure, by trying out various substitutions and uses three-word queries extensively, (s)he will succeed for

11 additional topics (#351, #353, #358, …). In other words, at least one of the three-word queries succeeds for 32 topics.

We justify the binary view of success shown in Table 6 by the fact that in real life:

- query sessions have a limited length
- after any query, success or failure may be considered
- success/failure regarding the session may depend on the history of the session, all the retrieved documents collected so far, etc.
- success/failure may not be a binary thing, e.g., the retrieved set of relevant documents may have value of various degrees

Above, we studied a more limited case where:

- sessions have a limited length
- each query within a session succeeds or fails
- the session ends successfully whenever a query succeeds
- the session fails if none of its queries succeeds
- the criterion for binary success is defined as follows: finding one highly relevant document is counted as success (P@5 = 0.2, 0.4, 0.6, 0.8, or 1.0) for any one particular query combination for the topic. Note that the binary success criterion can be defined in many other ways, e.g., as P@10 > 0, using liberal relevance threshold.

Last, we will show the traditional average precision interpretation of the effectiveness of the query combinations (Table 7).

Table 7. Non-interpolated average precision (%) for the various query combinations averaged over topics (N=38) (stringent relevance threshold, top-1000 documents retrieved).

				{}					
		11.2	7.2	0.8	1.3	1.3			
18.5	15.1	13.1	10.5	7.7	11.2	6.5	4.2	4.3	2.7
19.1	18.7	15.9	15.7	15.3	11.1	12.1	8.1	8.8	5.8
		21.1	20.3	17.0	16.4	11.6			
				17.9					

Table 7 presents the non-interpolated average precision results based on the top-1000 documents retrieved (stringent relevance threshold). Very short queries appear as inferior compared to the longer queries.

5. Discussion and Conclusions

The list of limitations of Cranfield-style experiments discussed in *The Turn* suggests that the effectiveness of IR methods and systems should be evaluated through several short queries, and assuming multiple-query topical sessions, because such an approach better corresponds to real life IR. We suggested in this paper that a graph-based simulation allows *retrospective analysis of the effectiveness of short-query sessions*. We assumed that a set of alternative queries is available for each topic, and the simulated user may try them in various combinations. The effects of word-level modifications in sessions may be considered systematically (e.g., one-word additions, deletions and substitutions, or more expensive operations) using the graph-based approach.

Note that the shortest queries in our experiment differ from utilizing, e.g., title queries of test collections. In the test data only three topics had a title field containing one word (#364: rabies; #392: robotics; #403: osteoporosis); for 19 topics the title field had two words, and for 19 topics three words. We experimented by trying out, e.g., several one-word queries for each topic. If we use P@5 > 0 as the success criterion (one highly relevant document required), in case of 15 topics (out of 38) success is reached by either the very first one-word query candidate ('A'), or the second ('B'), if the first one failed.

Our approach offers an instrument for comparing IR system performance when we assume input from users who behave by trying out one or more queries, as a sequence, but which may be very short, ambiguous, or both. The graph form allows presenting alternative query versions and considering their systematic modifications. By using a binary success criterion (e.g., P@5 > 0) we may investigate what kind of an IR system should be rewarded. For example, assume an IR system which is able to disambiguate query keys, cluster documents, and offer distinct interpretations for the query key (e.g., jaguar) – to offer one document as a representative for each cluster. The binary success criterion rewards this kind of system, because one correct interpretation in top-5 suffices for success but the system is not rewarded for finding more than one relevant documents (unless the threshold is raised). An IR system performing well – measured this way – is interesting from the user's point of view, because real searchers do use ambiguous words as queries – even as single words. Note that a set of alternative topical queries are needed because in real life the users consider keys from among several alternatives.

Peter Ingwersen [14] identified a phenomenon called the Label Effect. He wrote that searchers tend to act a bit at random, to be uncertain, and not to express everything they know. Instead, searchers express what they assume is enough and/or suitable to the human recipient and/or IR system. They compromise their statements under influence of the current and historic context and situation. In addition, the label effect means that searchers, even with well-defined knowledge of their information problem, tend to label their initial request for information verbally by means of very few (1-3) words or concepts. This description fits well what other studies [3] [6] tell about searcher behavior in the Web or intranets. It also closely matches the simple query session strategies that we propose to simulate in the present paper. In other words, we propose simulation of searching under the label effect.

We focused on retrieval situations where the searchers take their chances by repeatedly trying out short queries. We used a very limited set of query keys in our experiments. However, in the future IR test collections can be extended so that the facets of the test topics and their expressions are suggested by test searchers. Furthermore, the expressions of the facets in the relevant documents can be recognized. This kind of data could be used for more extensive graph-based session simulations. Our initial results indicated that even one-word queries often bring rapid success if they are considered as sequences. We suggest that the effectiveness of IR systems and methods should be compared, in test collections, from this perspective in the future.

Appendix

The five query words corresponding to A, B, C, D, E in Figure 1 are listed below for 41 topics. Due to lemmatization sometimes one user-given key produced more than one word. Due to the limited number of distinct search words given for some topics, some keywords are repeated. For topics #378, #414, and #437 no highly relevant documents exist in the recall base.

#351: petroleum, exploration, south, atlantic, falkland
#353: exploration, mine, antarctica, of, research
#355: remote, sense, ocean, radar, aperture
#358: alcohol blood, fatality, accident, drink drunk, drive
#360: drug, legalization, addiction, drug, drug
#362: realize, incident, smuggle, incident, gain
#364: rabies, cure, medication, confirm, confirm
#365: el, nino, flood, drought, warm
#372: native, american, casino, economic, autonomy
#373: encryption, equipment, export, concern, usa
#377: popular, cigar, smoke, night, room
#378: opposite, euro, reason, use, refuse
#384: build, space, station, moon, colonize
#385: hybrid, automobile, engine, gasoline non, engine
#387: radioactive, waste, permanent, handle, handle
#388: biological, organic, soil, use, enhancement
#392: future, robotics, computer, computer, application
#393: mercy, kill, support, euthanasia, euthanasia
#396: illness, asbestos, air, condition, control
#399: undersea, equipment, oceanographic, vessel, vessel
#400: amazon, rainforest, preserve, america, authority
#402: behavioral, generic, disorder, addiction, alcoholism
#403: elderly, bone, density, osteoporosis, osteoporosis
#405: cosmic, event, appear, unexpected, detect
#407: poach, impact, wildlife, preserve, preserve
#408: tropical, storm, casualty, damage, property
#410: schengen, agreement, border, control, europe
#414: sugar, cuba, import, trade, export
#415: golden, triangle, drug, production, asia
#416: gorge, project, cost, finish, three
#418: quilt, money, income, class, object
#420: carbon, monoxide, poison, poison, poison

#421: industrial, waste, disposal, management, storage
#427: uv, ultraviolet, light, eye, ocular
#428: decline, birth, rate, europe, europe
#431: robotic, technology, application, century, th
#437: deregulation, energy, electric, gas, customer
#440: child, labor, elimination, corporation, government
#442: hero, benefit, act, altruism, altruism
#445: clergy, woman, approval, church, country
#448: shipwreck, sea, weather, storm, ship

References

1. Ingwersen, P. and Järvelin, K. (2005) The Turn: Integration of Information Seeking and Retrieval in Context. Heidelberg, Springer, 2005.
2. Kekäläinen, J. and Järvelin, K. (2002) Evaluating information retrieval systems under the challenges of interaction and multi-dimensional dynamic relevance. In CoLIS4, 253-270.
3. Jansen, M. B. M., Spink, A., and Saracevic, T. (2000) Real Life, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web, Information Processing & Management, 36(2): 207-227.
4. Järvelin, K., Price, S. L., Delcambre, L. M. L., and Nielsen, M. L. (2008) Discounted Cumulated Gain Based Evaluation of Multiple-Query IR Sessions, in Proc. ECIR'08, 4-15.
5. Smith, C. L. and Kantor, P. B. (2008) User Adaptation: Good Results from Poor Systems, in Proc. ACM SIGIR'08, 147-154.
6. Stenmark, D. (2008) Identifying Clusters of User Behavior in Intranet Search Engine Log Files. Journal of the American Society for Information Science and Technology, 59(14): 2232-2243.
7. Turpin, A. and Hersh, W. (2001) Why Batch and User Evaluations Do Not Give the Same Results, in Proc. ACM SIGIR'01, 225-231.
8. Swanson, D. (1977) Information Retrieval as a Trial-and-Error Process. Library Quarterly, 47(2): 128-148.
9. Sanderson, M. (2008) Ambiguous Queries: Test Collections Need More Sense, in Proc. ACM SIGIR'08, 499-506.
10. Lorigo, L., Haridasan, M., Brynjarsdottir, H., Xia, L., Joachims, T., Gay, G., Granka, L., Pellacini, F., and Pan, B. (2008) Eye Tracking and Online Search: Lessons Learned and Challenges Ahead. Journal of the American Society for Information Science and Technology, 59(7): 1041-1052.
11. Sormunen, E. (2002) Liberal Relevance Criteria of TREC - Counting on Negligible Documents? In Proc. ACM SIGIR '02, 324-330.
12. Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. (2005) Accurately Interpreting Clickthrough Data as Implicit Feedback, in Proc. ACM SIGIR'05, 154-161.
13. Pirkola, A. and Keskustalo, H. (1999) The Effects of Translation Method, Conjunction, and Facet Structure on Concept-Based Cross-Language Queries. Finnish Information Studies 13, Tampere, 1999. 40 p.
14. Ingwersen, P. (1982) Search procedures in the library analyzed from the cognitive point of view. Journal of Documentation, 38(3): 165-191.